

2021

## Overcoming Team Boundaries in Agile Software Development

Scarlet Rahy

University of Salford, s.rahy@edu.salford.ac.uk

Julian Bass

University of Salford, j.bass@salford.ac.uk

Follow this and additional works at: <https://scholarworks.lib.csusb.edu/jitim>



Part of the [Communication Technology and New Media Commons](#), [Computer and Systems Architecture Commons](#), [Digital Communications and Networking Commons](#), [Information Literacy Commons](#), [Management Information Systems Commons](#), and the [Other Engineering Commons](#)

---

### Recommended Citation

Rahy, Scarlet and Bass, Julian (2021) "Overcoming Team Boundaries in Agile Software Development," *Journal of International Technology and Information Management*. Vol. 29 : Iss. 4 , Article 2.  
Available at: <https://scholarworks.lib.csusb.edu/jitim/vol29/iss4/2>

This Article is brought to you for free and open access by CSUSB ScholarWorks. It has been accepted for inclusion in *Journal of International Technology and Information Management* by an authorized editor of CSUSB ScholarWorks. For more information, please contact [scholarworks@csusb.edu](mailto:scholarworks@csusb.edu).

# Overcoming Team Boundaries in Agile Software Development

Scarlet Rahy  
(University of Salford)  
Julian Bass  
(University of Salford)

## ABSTRACT

*The usage of agile software development methods is increasing and so is the need for enhancing the collaboration between the different stakeholders. Thus, we chose to investigate the communication tools and challenges across the different boundaries and consequently deduce implications for practitioners. This research addresses inter-team communication by exploring the practitioners' perception on the different communication tools and the challenges faced at the three different boundaries, inter-team, team and customers, and geographically separated teams. We aim to enhance the productivity of software development through enhancing the communication between the different stakeholders. In this research, we use grounded theory approach to gather data from semi-structured open-ended interviews with practitioners in a geographically separated software development company. The findings observed three main inter-team communication means (Slack, Trello, face-to-face) used by practitioners, and how the preference differs among practitioners. This study also focuses on the different challenges faced, such as absence of communication during user story dependency, unclear customer requirements, and cultural differences, when implementing agile across different geographical locations. Our study discovers how team boundaries are overcome when team members adapt to other teams' preferred communication tool and enhance the boundary spanners' role. Furthermore, the study highlights how novice and mature teams require a transition time to adjust to the agile methods.*

**Keywords:** agile information system development, inter-team communication, agile team boundary, communication, agile methods, overcoming team boundaries

---

## INTRODUCTION

The creation of the Agile Manifesto in 2001 has changed the software development field through increasing customer involvement and creating business value (Dingsøyr, Nerur, Balijepally, & Moe, 2012). Agile software development is growing deeply within the same organization and horizontally spreading across different organizations (CollabNet & Version One, 2018). Furthermore, it has increasingly become an essential component to overcome competition; and with globalization, organizations expanded their development teams to different geographical locations (Herbsleb & Moitra, 2001). Studies suggest agile implementation as means to improve quality management systems (Krehbiel & Miller, 2018). Multi-cultural and geographically distributed software development models are becoming more common. Thus, rises the need to study the challenges and practices of information sharing in geographically distributed agile software development teams -Global IT, 2012) and the way agile implementation differs across the diverse geographical locations.

Methods of agile software development have emerged in the late 1990s as a response for the weakness of traditional methods. The agile methods are based on the values and philosophies developed in the Agile Manifesto (Beck et al., 2001), which promotes the continuous involvement of the customer and encourages feedback. The goal of agile application lies in adaptability, flexibility and responsiveness (Dyba & Dingsoyr, 2009). Agile uses experimentation and introspection as means to adapt to the constantly changing world (Dyba & Dingsoyr, 2009). This is possible through software development methods that include Lean Software Development (Poppendieck, 2003), Scrum, and Extreme Programming.

Agile teams cannot work in isolation, thus inter-team communication is a necessity to achieve the success of agile software development (Dingsøyr, Moe, Fægri, & Seim, 2018; Hoda, Noble, & Marshall, 2013; Santos, Goldman, & de Souza, 2015; Šmite, Moe, Šablis, & Wohlin, 2017). Self-organizing agile teams should sacrifice some level of autonomy (Bass & Haxby, 2019). The delivery of customer requirements needs to be the product of inter-team coordination (Bass & Haxby, 2019). Failure to achieve effective communication in the agile development process leads to failed patterns in delivery. A key challenge for communication among teams is building trust. Establishing trust is important all through the development process but is highly essential in the early stages of the process for example during kick-off meetings (Trainer & Redmiles, 2018). Such gaps in communication may lead to misleading or low-quality information and processing late work. This may cause the formation of false impression, usually negative, formed on other team members or across different teams. This results in low productivity in software development. Thus, inspecting communication means and ways to enhance them is a necessity especially in software development to increase productivity. We answer the call for further research on team communication and coordination in agile software development by (Dingsøyr et al., 2018). This study examines the means of communication used across the different teams in a software development company spread across two geographical locations: The Netherlands and Kenya. It also addresses the issue of communication across three different boundaries: inter-teams, teams and customer, and geographically separated teams; taking into consideration the difference in the challenges faced when implementing agile.

---

The paper highlights the challenges faced between the different stakeholders. The contribution of the paper is the discovery of the different mechanisms used to overcome the challenges at the level of the three studied boundaries. This paper also detects the communication improvement when different preferred communication mechanism between entities are adopted.

We designed this case study in particular to learn more about information flows in agile teams. In order to understand the above, we will study the agile teams and their choice of communication tools in the chosen company. Thus, we need to answer the following research questions:

RQ1: What inter-team communication tools do practitioners in different roles advocate in our study? And

RQ2: What communication bottlenecks do practitioners face at the three studied boundaries, inter-teams, teams and customer, and geographically separated teams?

In addressing the research questions, the paper contributes through highlighting the mechanisms used to overcome challenges at the different boundaries. These mechanisms include adapting to different communication means, enhancing the role of the boundary spanners and benefiting from different geographically separated teams' experience.

The rest of the paper is structured as follows. In the next section, an overview of the literature review is presented and includes an overview of agile development, inter-team information sharing, trust, and boundaries in agile software development. Then the methodology is presented. The study addresses the above research questions in a qualitative study that uses grounded theory methodology. Nine practitioners were interviewed from an international software development organization. Data collected was then analyzed using open coding and constant comparison technique. Then, the findings of this study are introduced and organized into two main parts. The first part discusses the inter-team tools that are used for communication such as team messaging tools, face-to-face, and virtual Kanban boards. The second part discusses the communication boundaries between teams inside the company, between teams and customers, and between teams that are located in separate geographical locations. In addition, the latter part observes the differences in challenges faced when implementing agile in different geographical locations. Further the discussion section analyzes the findings and presents a limitations section along with implication for practice. Finally, the conclusion is presented followed by the list of references.

## RELATED WORK

Self-organizing teams are seen as the spirit of Agile, focusing on social and human aspect of software engineering. The Agile Manifesto encourages effective communication through motivating and supporting such kind of teams (Beck et al., 2001). Communication is essential for the implementation of agile practices, for instance, the daily stand-up meetings with 90% usage, followed by sprint planning with 88% usage, and thirdly by retrospectives with 85% usage (CollabNet & Version One, 2018). The daily stand-up meetings occur daily between the team members in a predetermined space and time to discuss what has been done, what is to be done and impediments encountered, if any (Stray, Sjøberg, & Dybå,

2016). Sprint planning occurs when team members gather to share the details on user stories' complexity, utility, and dependency (Boschetti, Golfarelli, Rizzi, & Turricchia, 2014). Retrospectives are devoted to the improvement of the agile software development process and adaptation to changes that arise (Jovanovic, Mesquida, & Mas, 2015).

Inter-team collaboration and coordination are bases for agile software development (Santos et al., 2015). All team members are required to have a dynamic behavior adaptable to the customer's requirements since customers are major players in the agile software development journey (Praby & Roland, 2016). The decision relating to the choice of agile methodologies is highly influenced by the customer (Rajagopalan & Mathew, 2016). The identification and prioritization of customer requirements is however conflicting (Bass, 2015; Praby & Roland, 2016). Customer feedback is considered by some team members as personal criticism and an offense. Thus, team members focus on self-defense rather than work adjustment and adaptation (Hoda et al., 2013).

Communication, feedback, coordination and collaboration are key characteristics for a self-organizing agile team (Drury, Conboy, & Power, 2012). Team members should encourage information dissemination, even in the least favorable information collection strategies, for instance formal reports (Bass, Allison, & Banerjee, 2013). In addition, the Agile Manifesto grants all team members the opportunity to participate in the decision-making process (Beck et al., 2001) through striving towards consensus-based decisions with opinions from all team members being valued (Hoda et al., 2013). This creates obstacles since some team members show reluctance in decision-making and may consider it as a burden rather than a privilege and rely on the scrum master for taking decisions (Drury et al., 2012). In such cases, scrum masters tend to take one of these approaches: taking the decision and informing the team members, in turn contradicting an agile principle, encouraging team members and waiting for their response or using decision-making support systems to aid in the process (Anika Rani, Shahper Vodanovish, David Sundaram, 2015).

Inter-team information sharing has been identified as a key factor in agile software implementation by several researchers. Agile software development requires the transfer of information throughout several stages, which may lead to miscommunication and extensive communication paths (Chau & Maurer, 2004). To compensate this, agile development uses cross functional teams and encourages open and direct communication. Dingsoyr et al. indicated that there are a number of areas to achieve knowledge sharing and coordination in teams, and these change over time (Dingsøy et al., 2018). Santos and Goldman developed a theoretical model that enhances the effectiveness of inter-team knowledge sharing through organizational conditions, such as team integration and agile methods adaptation, as well as stimuli that include motivation toward a common goal and incentives (Santos et al., 2015). Šmite et al. emphasized the importance of establishing a networking culture to enhance cross team interaction as a driver for better performance (Šmite et al., 2017). Networking behavior is affected negatively by factors including complexity and unfamiliarity of the tasks, frequent process changes and cultural differences (Šmite et al., 2017).

It is a necessity that the different stakeholders, in an agile implementing environment, trust and respect each other (McHugh, Conboy, & Lang, 2011). Trust cannot be created; rather it is a process that develops and grows over time through repeated interaction and shared experience (Trainer & Redmiles, 2018). Agile methods, such as pair programming and collective code ownership, require the presence of trust. Consequently, the communication practices in agile enhance and work on building this trust. These practices include retrospectives, daily stand-ups, and iteration planning. Open communication with the team, frequent knowledge sharing across teams, and honest feedback in retrospectives are all factors that have a direct impact on trust between team members and across teams (McHugh et al., 2011). Awareness on each team members' collaboration leads to accurate attributions

and trustworthiness of each member (Trainer & Redmiles, 2018).

One of the four main themes of concern in agile implementation is inter-team coordination or dependency (Dingsøy et al., 2014). Interdependencies between the different tasks of software development may be difficult to organize or prioritize (Okhuysen & Bechky, 2009). The number of teams and developers involved along with the workload and dependencies make team coordination and networking behavior essential to achieve project success (Licorish & MacDonell, 2014; Šmite et al., 2017). With the increase in dependent tasks, communication about mutual obligations increase. Here, team members are faced with different perceptions that might create a possible decline in trust (Moe & Šmite, 2007). In addition, communication barriers rise with mutual mistrust among the different roles (Ghobadi & Mathiassen, 2016). Such dependencies and criticisms can be facilitated by the presence of a boundary spanner (Šmite et al., 2017).

The nature of agile software development produces boundaries between different groups such as, inter-team, team and customer, and product owner and developer. Thus communication facilitators, such as the boundary spanners, rose to decrease the gap between these groups (Boden, Avram, Bannon, & Wulf, 2009; Šmite et al., 2017). Boundary spanners act as coordinators who provide a source of information, a target for feedback (Šmite, Moe, Šāblis, & Wohlin, 2017), a mediator between different teams (Yoo, 2008) and a socio-material assemblage (Doolin, 2012). In addition to facilitating internal communication, boundary spanners help form organizational identities (Gal, Lyytinen, & Yoo, 2008).

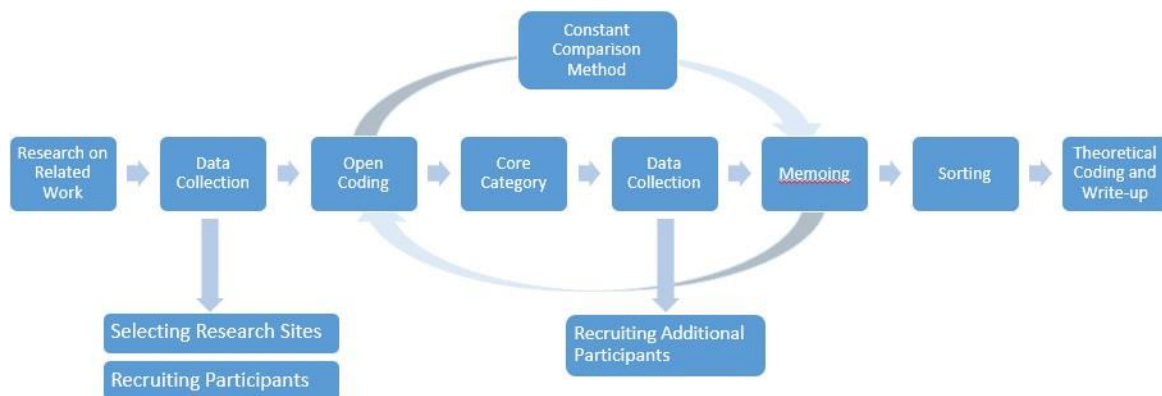
The global software development research tracks three main areas in distributed collaboration: coordination, temporality, and communication. Our study focuses on communication and the importance of developing a collaborative nature between geographically separated teams and among the members of the same team. Extensive research on agile software development principles and practices is known (Adolph, Hall, & Kruchten, 2011; Alahyari, Berntsson Svensson, & Gorschek, 2017; Bass, 2015; Hoda, Noble, & Marshall, 2012; Petersen, Roos, Nyström, & Runeson, 2014), but less is done on inter-team communication. Inter-team knowledge sharing in agile software development is still in the rise (Dingsøy et al., 2018; Santos et al., 2015; Šmite et al., 2017), and is identified as an important topic in research (Dingsøy et al., 2018). Practices that are applied specifically for knowledge sharing in agile software development are still under study. Previous research has investigated inter-team communication tools but our research addresses the information sharing challenges at the team boundaries and investigates how to overcome them.

## MATERIALS AND METHODS

A qualitative research methodology is adopted for this study and used as a basis for the grounded theory approach. We employed the grounded theory, a qualitative research methodology, since it allows us to address the complexity of human behavior in software development context (Seaman, 1999). Grounded theory allows us to uncover new concepts in terms of inter-team communication tools used and the communication at the different boundaries. It allows the researcher to uncover new concepts from the data and generate theories rather than creating a clear research hypothesis (Hoda & Noble, 2017). This theory generation allows us to uncover bottlenecks faced by participants when it comes to information flow within the team and across different boundaries and their respective

resolutions. The primary concepts are identified through common themes and patterns that emerge from the analysis of the dataset collected.

We chose grounded theory to be our research methodology for various reasons. First, grounded theory has been used to study agile software development by numerous researchers (Bass, 2015; Hoda, Noble, & Marshall, 2010; Patton, 2002). Second, there exists a high synergy between our topic, agile methodology, which focuses on people interaction and our research method, grounded theory, which uses qualitative research methods to study social interactions. Third, grounded theory aids in answering our research questions that revolve around understanding the practitioner's perception rather than theory analysis or theory extension. By using open-ended research questions asked in the interviews, the topics will be generated by the participants. In the following subsections, we describe the key components of Grounded Theory, open coding, constant comparison method, core category, selective coding, memoing, sorting, theoretical coding, and write-up, following the model in figure 1 (Hoda et al., 2010).



**Figure 1. Overview of Grounded Theory Method - Adapted from Hoda et al. (2010)**

The three pillars of qualitative research include open ended interviews, direct observation, and written communication (Patton, 2002). In this study, semi-structured open ended interviews were used to collect data (Vaivio, 2012). The unit of analysis are employees and product owners at this study's research site. In the first phase, we conducted an exploratory pilot study to refine and insert alterations in the questions and to enhance our knowledge with this type of research. In the second phase, we performed a deductive synthesis of a series of interviews to analyze the data.

### *Data collection*

This section describes the research sites, process of recruiting participants and conducting interviews. There were 9 participants in our study working in an international agile software development companies. This company develops business solutions for agile companies, non-agile companies, and governmental departments using agile software development

techniques and conducts their business administration using enterprise agile. The set of the open-ended questions in the interview were designed to address the research questions in our study. The following are examples of the main categories and one of the corresponding open-ended interview questions used to gather data: (a) inter-team communication tools: “How do you communicate with the people outside the team and at what frequency?”; (b) communication at the boundaries: “How do dependencies between teams affect the flow of information and work?”.

### ***Research Sites***

Data was collected and analyzed from an internal agile software development company that provides services for large software projects. The company uses agile to provide business solutions and create custom software. It has a main office in The Hague, The Netherlands (50 employees) and a partner office in Nairobi, Kenya (15 employees). The company was founded in 2005 and has a diverse set of clients spread across Europe and Africa. It was chosen according to the snowball sampling technique; academic contact eased the connection.

In the second phase, the professional contact provided access and organized a schedule to interview the participants.

### ***Recruiting Participants***

Participants were interviewed with different responsibilities and spread across different locations in the company. An overview of the participants’ role and responsibility, location, and years in company are shown in Table 1. Employees interviewed have been in the company for a range of 6 months to 3 years except for the director, who has been in the company since 2005. The data collected was obtained from semi-structured open-ended questions. The length of the interviews ranged from 45 minutes to an hour and 15 minute interview discussing the participants’ experience on information-sharing in an agile environment. The most effective way to optimize the data collected from interviews is to record and transcribe data manually (Adolph et al., 2011). All interviews were recorded after obtaining the practitioners’ consent, and to ensure that no data is lost and that we concentrate on the conversation rather than jotting down notes. Then interviews were transcribed manually since it ensures correct transcription and reminds the interviewer of the social and emotional aspects that occurred during the interview (Vaivio, 2012). The conducted interviews followed an open-ended questions guide that enabled participants to raise any issue that came up.

**Table 1. Participants' Roles, Responsibility, Location, and Years in Company**

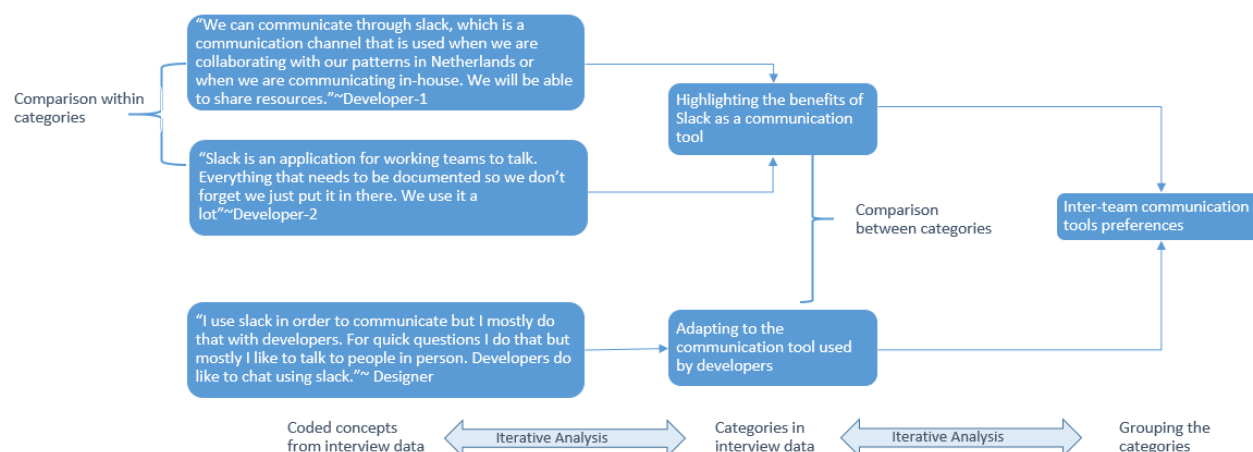
<b>Participant</b>	<b>Role and Responsibility</b>	<b>Location</b>	<b>Years in company</b>
Director-1	Director of the Company	Kenya	4 years
Director-2	Director of the Company and Product Owner	Netherlands / Kenya	7 years



Technical Lead	Technical Lead and Product Owner	Netherlands / Kenya	4 years
Public Relations (PR)	Public Relations Manager	Netherlands	6 months
Sales	Sales Coördinator	Netherlands	3 years
Designer	Designer and part of the Public Relations team	Netherlands	2 years
Developer-1	Scrum Master and Developer	Kenya	2 years
Developer-2	Front End Developer	Netherlands	3 years
Human Resource (HR)	Human Resources Manager	Netherlands	6 months

### Data analysis

The following sections describe the Grounded theory data analysis procedures that were conducted including open-coding and constant comparison. From the analyzed data, interview concepts arise. These concepts are then grouped into categories which then form the main categories. An example of the above process is shown in figure 2. This figure examines one example of comparison of quotes between and within categories of inter-team communication tools preferences.



**Figure 2. Iterative analysis example**

---

## *Open Coding*

All transcribed data was imported to an analyzing tool (Nvivo 11). Open coding includes the analysis of data and exclusion of prior judgment to produce the maximum number of concepts and key points. All interviews were coded, to derive categories on a high levels of abstraction, along with patterns of behavior (Adolph et al., 2011). Then, each key point is assigned a code titled with a phrase summarizing it.

Line-by-line open coding approach was used on the transcribed interviews, which is more effective and useful than word-by-word coding approach. When coding line-by-line, data can be inspected and a special incident can be found in a word or line, or through several lines (Adolph et al., 2011). This coding process was applied through highlighting the selected sentences using the Nodes option in Nvivo 11. In order to ensure consistency, the collection and analysis of data was performed by the same individual. Using the constant comparison method, each interview was reviewed more than once to confirm that no data is left unnoticed. This process reached a halt when no new categories were created, and theoretical saturation was reached.

## *Constant Comparison Method*

Codes from each interview were compared from codes arising from the same and other interviews. This constant comparison technique enabled the grouping of codes that constitute concepts (a higher level of abstraction) which were then grouped, using additional constant comparison, into categories (an even higher level of abstraction) that were then coded.

## *Core Category and Selective Coding*

After open coding, a core category emerged from the data collected. The core category is chosen according to its centrality and relation to other categories that emerged, in addition to its frequent reoccurrence in the collected data. The core category that emerged revolves around communication tools and boundaries. After selecting our core category, we moved to selective coding which involves limiting the coding to data that relates to the core category.

## *Memoing and Sorting*

The next step was the ongoing process of writing memos under the grounded theory process. Memoing is considered to be the ‘backbone’ of theory generation (Glasser, 1978). Memos were first written in their draft version, to guarantee that they are written in the “passion of the movement” (Adolph et al., 2011), and then were visited and written in a formal manner, to ensure clarity through using correct and revised English. Quotes from the interviews were used as evidence in the writing of memos.

Next, the written memos were sorted on the conceptual level to relate the created categories to the core one. This was followed by relationship creation between memos through drawing out the relationships using a paper and a pen (Hoda et al., 2012).

### *Theoretical Coding and Write-up*

Consequently, theoretical codes were highlighted from the emerged concepts and categories. Relationships were analyzed between the different categories and their respective properties to reach our theory. The writing-up of the theory came as a result from the sorting and theoretical coding. Our study revolves around communication tools used at the inter-team boundary and the communication challenges faced at the three studied boundaries, inter-team, team and customer, and geographically separated teams.

## RESULTS

In order to address research questions 1 and 2 represented in the section 1, we interviewed practitioners, established core categories, and formed the below results backed-up by extracts from these interviews.

### *Inter-team Communication Tools*

During interviews, practitioners were asked about communication in general. For example, such questions were asked: “How is the informal communication transferred from one actor to another? And how frequent?”. All questions regarding communication can be found in the interview guide under ‘Communication’ section in Appendix A. These questions help us determine the communication tools used by the practitioners in our case study.

### *Collective Record of Communication Tools Used*

Different communication forms and tools were revealed by practitioners during interviews including team messaging tools, face-to-face communication, emails, video conferencing tools, and Kanban boards. The availability of communication tools and technologies is at the same level in both company premises in Kenya and The Netherlands, since they are part of the same company. Below is table 2 that includes all practitioners’ responses on the communication tools used. For each practitioner, a quote describing the communication tools is extracted and the respective communication tools are indicated.

**Table 2. Communication Tools used by Practitioners**

Practitioner	Quote	Messaging Tools	Face-to-face	Emails	Video Conferencing	Kanban Boards
Director 2	<i>“The most is talking, then messaging, then passing user stories through Trello boards is less frequent but they have bigger impact.”</i>	x	x			x

HR Manager	<i>"A lot passes through email. We use Trello boards and slack as internal chat. I think these are the three main communication forms. And of course face to face."</i>	X	X	X		X
Designer	<i>"We start first with the briefing at the sprint planning that is made by user stories that are put on Trello. ...Most of the communication is just speaking together or on Slack... we use email if the person is not in the office."</i>	X	X	X		X
Developer 1	<i>"We may communicate through Slack or Trello. But we are in an open office environment, so we are able to talk to each other."</i>	X	X			X
Developer 2	<i>"We do a lot of communication via Slack. Also we use Trello to track user stories when we develop them in the meeting."</i>	X	X			X
Technical Lead	<i>"We use Trello and this is where we set all the user stories for the teams. We use slack. And of course communicate face-to-face daily."</i>	X	X			X
Director 1	<i>"We use Slack for written chatting, Google hangout for video conferencing, and the digital scrum board Trello for communicating the stories. The developers that are physically together in the office they just talk."</i>	X	X		X	X
PR	<i>"We can chat on slack if it is internally...I can simply talk to the people... I put my stories on Trello."</i>	X	X			X
Sales	<i>"I use Slack...Rarely via email but it is not a very big office so I just walk to them and ask them a question."</i>	X	X	X		

From the above table, we are able to recognize the three most important communication platforms used by practitioners. Communication and coordination enhancement between teams is mostly achieved through team messaging tools (or social networking platforms), face-to-face, and virtual Kanban boards. In the below sections, we discuss the above mentioned tools since they are the ones recurrently mentioned by the practitioners.

### *Team Messaging Tools (Social networking platform)*

Enabling coordination and communication is a key factor of success for agile implementation. In this company, Slack is used as a social networking platform for employees to communicate together (Slack, version 3.3.1). This application serves as a digital workspace for daily communication and knowledge facilitation. The interviews conducted highlight two major points of view on Slack usage. Developers were characterized as advocates for regular Slack usage, while designers, sales, public relations, and human resources teams were reluctant towards Slack usage.

The developers advocated the daily usage of Slack as means to communicate with members of the same teams, members of other teams, and teams located in different geographical locations.

Developers highlighted the numerous benefits of slacks. First, Slack allows open communication between members of the same team, across teams, and within the company as a whole. According to Developer-1:

*"In Slack we are able to communicate to a group. A group can keep up with the communication and know what is going on in different aspects. Also they can chip in if they feel there is something they can input on any matter."*

In addition, the Technical Lead highlighted one of Slack's benefits as a platform to request and receive support when needed: "...we have channels for the teams on their own and we have channels for the project we have a channel were people can ask for help".

Slack also has features which aid in simplifying and facilitating the communication process such as notifications, tags and pins. Developer-1 noted that "If we have a group that is specific for a certain project, I am able to tag the members of the team to draw their attention to something... everyone will get notified as soon as I post something". Moreover, Developer-2 revealed that Slack may be used when team members are absent for a certain period of time: "If we need to remember something for the next day we pin it on Slack so we can see it the next day", and the Technical Lead agreed. Developer-1 highlighted the importance of the tag and notification option in Slack: "Slack has a notification feature that allows everyone to get notified as soon as I post something. It allows also for tagging. For instance I can tag specific people or the whole team."

In addition, developers use Slack for saving documents, customer requests, and conversations. Developer-2 indicated how slack can be used as a memory box: "Everything that needs to be documented so we don't forget we just put it in [Slack]".

Contrary to developers, public relations, human resources, sales and designers were reluctant towards the usage of Slack. Non-developers tend to use Slack solemnly as means to communicate with the developers. The Designer said: "I use Slack in order to communicate but I mostly do that with developers". The Sales person also showed discomfort with the usage of Slack: "I use Slack but not a lot...the only people I know who use Slack are developers". Similarly, the HR manager, indicated: "Slack is my least favorite form of communication." while the Technical Lead noted that "It is more difficult for non-developers to actually express by text what they mean". The non-developers have the tendency to use face-to-face verbal communication which decreases the usage of Slack to a minimum. This

results in a trade-off between the availability of information for all employees at any time and the benefits of verbal communication.

Developers argue that verbal communication reaches solemnly the involved actors, while information shared on Slack is spread to all concerned actors.

Furthermore, Slack can be used to ask and answer small and prompt questions. If a developer has a specific question about a certain feature, such as color or font, this is best transferred through Slack.

The Designer said: *“They [developers] always have short questions, like what is the color of this design, how big this should be... and many questions like that during the development process”*.

### ***Face-to-face***

Collaboration, creativity and team building is enhanced through face-to-face communication in the workplace. One of the main advantages of face-to-face communication is the live feedback through body language, facial expressions, tone, reactions and feelings. All the employees use face-to-face communication, although, the preference differs from one employee to the other. Director-2 said: *“When most people use a document we try to use an email. When most people use email we try to use chat message. When most people chat we try to talk to each other”*. Face-to-face communication is used during agile ceremonies such as daily stand-ups, demos and retrospectives, as well as regularly daily.

Contrary to developers, the PR, Sales, HR and designers consider face-to-face to be their preferred form of communication and tend to use it frequently. The Designer indicated that face-to-face communication is the initiation of any project design: *“Without face-to-face communication I cannot start with my design.”* The HR manager seconds that by stating: *“Sometimes you need important information, and this is best said using face-to-face communication.”* Director-1 highlighted the significance of body language in communication: *“I prefer face-to-face physical communication because communication is not just text, it is also body language and tone facial expression which I think you lose most of if you only type, text or email.”* The Designer continues to stress on the importance of face-to-face in understanding the requirements: *“I think the person-to-person communication is really vital for understanding the other person and what I need to design.”*

Due to geographically separated teams, Skype becomes a regular mean for communication. Challenges rise especially with weak network signal, thus making remote communication between Kenya and The Netherlands inexpedient. The Designer indicated that *“sometimes in Kenya they are a bit slower and they have trouble with internet connection a lot. Sometimes the sprint would be affected by that”*; Developer-2 also indicated: *“We had a lot of stand-ups with them [Kenya office] that we had to cut short and continue later on because the connection was gone.”*

### ***Virtual Kanban Board***

Virtual Kanban boards are tools designed to visualize the current work in progress of all teams in agile software development. Employees recognized the importance of the virtual Kanban board Trello (Trello, version 2.8.3). Trello boards aid in managing the sprint process

through showing the sprint backlog list, identifying the work in progress, indicating the accomplished tasks and limiting work overload. The Technical Lead indicated how Trello boards reduce the number of regular meetings through transparently updating the progress of each team member. The Technical Lead said: *“Trello keeps an overview of all the project, sprints, and teams at the same time... With scrum boards in the rooms, one can see what the status is without too much hassle.”*

Moreover, Trello boards allow team members to check the work status of different teams by simply examining the board of the respective team. Trello boards highlight the status of user stories. The Technical Lead said: *“we use Trello and this is where we set all the user stories for the teams...when you pass by you can see straight away what the status is of the sprint”*. The Designer stated how convenient and practical Kanban boards are: *“The Trello board is more convenient to keep each other up to date and also as a reminder of how much we need to do. It is more like a practical thing.”*

### ***Communication at the Boundaries***

In the previous section, we indicated the difference in preferred form of communication between the teams. Alongside other factors, this causes communication complications at the boundaries between teams, product owners, scrum master, and customers.

### ***Inter-team Boundaries***

With the increase of time pressure and tight deadlines in the workplace, team members focus on completing their own user stories and tend to neglect requests from members of other teams. Consequently, they tend to prioritize the tasks given by their own team leader or scrum master and aim to achieve them, while postponing tasks or requests given by other team leaders. The Sales describes an incident that occurred with one of the developers under that context:

*“She started working at the customer but the screening wasn’t completed yet and she needed to hand a document. I emailed her, called and she wasn’t responding for several days and that was really frustrating. It took three days to complete that”.*

An additional issue is the lack of collaboration and continuous update among teams. Teams that are facing setbacks assume that other teams are aware of their situation thus do not feel compelled to explain their status. Developer-2 said: *“Teams are like, we know so the rest knows it as well. They are just assuming”*. The Technical Lead affirms the above by stating: *“They [team members] assume that if they mention that something did not go well then you know exactly what they are talking about even if they do not mention details.”* When such cases are repeated, trust between teams starts to decline. This causes significant problems in the performance of teams leading to poor performance of the overall project. As a result, the role of scrum master in inter-team communication is needed. The scrum master not only facilitates direct communication between the product owner and the team but also exchanges knowledge to and from members in and outside their team. Developer-2

---

describes an incident where team B was dependent on team A's successful sprint completion. Unfortunately, team A failed to complete their sprint and didn't communicate this through team B, who in turn couldn't initiate their own sprint. Developer said: *"If the scrum master had told the other team, they could have done a new sprint planning. Now they were just waiting and two teams were set back by that"*.

*Absence of Inter-team Communication In The Presence Of Dependencies.* The interconnection of sprints causes high level of dependencies. A sprint for Team A may be a prerequisite for the completion of the Team B's. Reliability between sprints may hinder the process, especially if the work done by Team A's sprint was incomplete or needed rework. An unsatisfactory sprint may contain or block the work flow. Developer-2 explained a similar case:

*"In the last sprint there was a team who picked up a bit too much, they underestimated their user stories and another team was dependent on what they were supposed to make, but they weren't informed of the delay. So they were waiting for their sprint to start for some time."*

The design team and the development team have encountered such cases numerous times. Since the developing team is highly dependent on the work of the design team, and since the sprint of the design team are highly dependent on the customer's approval, the developing team have experienced incidences where they had idle time. The Designer described: *"If something goes wrong in my sprint then the developers don't have anything to do"*.

The Technical Lead suggested that: *"The delay can be caused by miscommunication"*. Such cases lead to the obstruction of the work flow and may cause the delay of sprints, thus creating an idle time which could have been avoided if communication had occurred. Developer-2 said: *"So if they had told them [about the unfinished work] they could have done a new sprint planning, plan different stuff and finish other tasks"*.

### ***Team and Customer Boundaries.***

Communication with customers through agencies, dealing with unclear customer requirements, reaction to customer feedback, and collaborating with non-agile companies are issues that rise at the team and customer boundary.

*Communicating with Customers through Agencies.* Agile software development integrates customer feedback through its practices. In certain cases, a third party acts as a communicator between the company and the customer; consequently, communication is faced with interruptions. The Technical Lead said: *"when the development is through an agency, the communication is more difficult"*. First, knowledge sharing becomes difficult and unprecise, leading to frustration of team members. Second, restrictions and limitations that accompany the involvement of an agency hamper the team members' performance and enthusiasm. The Technical Lead said: *"This becomes time consuming. This is something that we don't have control over. It is somehow frustrating"*.



*Unclear Customer Requirement.* Customer involvement and input happens all through the agile process. Lack of clarity in customer requirements leads to unclear product backlogs, causing time delays in delivery. The PR manager said: *“The lack of information of the project we are doing is a factor that will negatively affect delivery time”*. The team members’ experience in formulating and asking the accurate questions at the beginning of the project prior to forming the product backlog is vital.

The Designer said: *“I ask a lot of questions that make the story clear and make a proper sprint planning out of it”*. Director-1 highlighted the importance of the scrum master’s role in clarifying the customer’s requirements: *“The scrum master can clarify the story or feature with the customer and product owner and communicate that to the team”*.

Unfortunately, sometimes the team members or the scrum master base their product backlog solemnly on customer’s briefing and fail to investigate further. Director-2 said: *“I think the most common negative effect on the workload is the lack of clarity and understanding in what is required”*. The designer also exclaimed: *“Sometimes, one can really come of track if you don’t understand the idea”*. Some clients might not be also involved in the process. Director-1 said: *“The clients give the requirements, but they don’t get involved so much”*.

*Reaction to Customer’s Feedback.* The success of the team’s demo to the customer and the following retrospective are related to the customer’s approval of the sprint. Some team members process the customer’s feedback during the demo personally and turn the retrospective into a session to discuss solemnly the demo instead of the sprint as a whole. Director-2 indicated: *“In our last demo we had problems with the customer that resulted in negative feedback, and then in the retro everybody put negative improvement “stickies” related to that particular incident”*. Some of the employees become directly affected by the customer feedback presented in the demo, which stretches further in affecting not only the retrospective but also the performance in the coming day. The Technical Lead indicates that: *“It depends if they get affected with the customer’s feedback”*. Developer-2 explained how negative feedback is dispiriting yet is essential to reach the customer’s requirements: *“It usually a bit demoralizing to start over again. But you have to start over and make sure that this is what they want next time.”* As the Designer said: *“It [customer feedback] can discourage you especially. I think as a designer you know the feedback will always come up, and it is never the way you thought it would be”*. Developer-2 talked as well about a recovery period: *“Usually the demo is on a Thursday and Friday is usually a personal sprint day, while Monday you can just start over. Friday would be like a day to recover”*.

On the other hand, some employees understand that continuous customer involvement is core in agile software development, and customers tend to change or refine their requirements frequently. Developer-1 indicated: *“The scrum agile process is actually about making the software based on the customer’s demand”*.

*Communicating with Non-Agile Companies.* Not all companies and customers implement agile. Director 2 describes a project of a highly regulated nature that requires setting milestones and documentation to develop a business strategy for a government department: *“We have a project with the German Government, which is completely non agile. The project description are non-agile. The documentation is non-agile and the terminology is non-agile”*. This means that some of the agile software development practices will not be completed in the same pace as with an agile implementing customer. Such customers will act as an obstacle towards the punctual completion of the project, since giving regular feedback is not implemented in their system.

Director 1 said:

*“Some of the features require another party to give us some feedback before we do or another company to do something. That is where things start failing because other parties or other companies don’t work the same agile manner we work. So, they became an obstacle or a bottle neck in our agile process, and you know you cannot push them too much so you can only hope they get back to you as soon as possible as not to become an embedment to the scrum process”*

In such case, being fully committed to implement agile, the company works on incorporating the non-agile requirements into the agile process through user stories. Director 2 explains: *“We just take an agile approach when dealing with non-agile.”*

### ***Geographically Separated Inter-Team Boundary in Relation to Agile Implementation.***

The experience in agile software development implementation differs from one geographical location and person to another. Director 2 expressed his view on agile implementation: *“Agile may be applicable to every culture, but within each culture it is not applicable to all people”*. The ability to implement agile depended on the communication effectiveness and software development background in both The Netherlands and Kenya.

*Struggles to implement agile in The Netherlands.* Employees with minor or no agile implementation experience find it difficult and time consuming to grasp the agile way of thinking. A PR employee with two months background in implementing agile finds it demanding to communicate and use the agile concepts: *“It is hard on me to adapt to the words used, thus it is hard for me to communicate”*. In addition, an HR employee, with 6 months of agile implementation experience, said: *“It is still tough to get my head about it, because it is a different mind-set”*. The HR employee also pinpointed that it is harder to implement agile in operational work as opposed to software development:

*“There is always going to be a discrepancy, because I am not building a software. It is different when you are applying agile to an administrative process. Some things are possible for certain projects, but there is also a lot of operational work, HR or finance, and I am not sure how you can change that into agile”*.

On the other hand, employees who have been implementing agile for a longer period highlighted several times the benefits of agile, and how communication increases its effectiveness. Director 1, with 3 years of experience in agile, said: *“Agile ensures that the management does not interfere in the project management, and the team is fully self-organizing and self-sufficient in terms of having what is needed to deliver that project”*. Also, the Sales, with 3 years’ experience in agile, said: *“We have user stories, it is clearer what we need to do. I think with agile and scrum it is more clear what needs to be done”*. At the same time, these employees recognize the difficulty in grasping the agile concept and implementing it.

Director 2 indicated: *“The rules of scrum are very simple. So I say, you can explain scrum and agile to somebody in half a day. But to really understand it and do it takes years”*. Moreover, the Technical Lead said: *“In the beginning we had many struggles to do it all over and over again. With some people it did not work at all, with some people it did, and for every person it is difficult how they pick up agile”*.

Director 2, said: *“Our experience is that some people are naturally good at scrum or agile”*. On the other hand, other employees found it hard to implement agile especially if they had previous work experience in a different field. The ability and will of every employee to forget any preconceived and practiced ideas about project management and start implementing agile differ. The technical lead highlighted: *“One learns how to do certain projects in a certain way, and if one wants to implement agile they need to unlearn things”*. In the Netherlands, employees have to unlearn preconceived ideas on project management and software development that are non-agile. The technical lead, who is based in The Netherlands, said:

*“Here, in The Netherlands, you learn how to do certain projects in a certain way and if you then are implementing agile you need to unlearn thing”*.

One specific example was given by Director 2 on one of the employees who came from a government sector which is characterized by long term planning, bureaucracy and perfecting both the value- and non-value-adding activities. The director 1 explained: *“So at the moment we got one person who is clearly struggling with the agile thing and she has come with a background in government which is quite old fashion bureaucracy [...] making a mental map that stretches into the future. And when she works on something, she assumes what things might be, and in agile you cannot do that. So we have a problem making people unlearn”*. *Struggles to implement agile in Kenya*. At the initial agile implementation period in Kenya, the employees at The Netherlands instructed their Kenyan partners on agile implementation. The process was difficult to grasp at first. Developer-1, based in Kenya, said: *“They were able to catch our weaknesses and give us a scrum agile implementation of a solution for the problems”*. The Kenyan partners were able to learn the agile process with a fresh mind-set. The process was a mutual benefit, where the employees at The Netherlands learned and enhanced the agile process back home while directing the Kenyans. The Technical Lead said: *“After every sprint in Kenya we learned new things and restarted the game”*. This mutual learning process affected the internal communication in The Dutch office with every practice of long-distance communication with the Kenyan one. Developer-2, having direct contact with the Kenyans, expressed:

*“If you managed to communicate with people in Kenya every day for 10 sprints in a row, then it is easier and becomes second nature to talk with people around you as well. Instead of calling them kilometers away you just walk the 5 meters to the next room and you talk with them”*.

This improvement and communication enhancement was also portrayed to employees in The Netherlands who weren't in direct contact with the Kenyan counterparts. The Technical Lead said: *“The whole company learned a lot about communication especially in the least two weeks our communication improved immensely”*; Director-2 expressed the same.

---

In addition, the office in Netherlands updated the way Trello boards were designed following a method created by the Kenyan partners. The Technical Lead said:

*“In Kenya, they came up with some ways to organize their Trello board for their office team and management team. We adapted and then changed that for a more suitable way for our office”.*

The third change was portrayed in the improvement of user story-writing. The Kenyans required detailed explanation in order to understand the work that needed to be done, and user stories were hence written with extensive description and detail. This became a habit in The Netherlands too, since it clarified more the user stories and decreased the number of questions asked later. The Technical Lead explained:

*“Because we added extra description in Kenya, we found out that we need to give a little more information that they can look up afterwards instead of telling that in the sprint planning next to the user story”.*







## DISCUSSION




This study responds to the call for further research on inter-team communication (Dingsøyr et al., 2018). We have described in the Findings section the inter-team communication and the corresponding used methods, along with the difficulties that rise at the boundaries between different actors and across diverse locations. This section discusses and analyses the case study and addresses the two research questions. In order to address research questions 1 and 2 presented in the section 1, we analyzed the results of the findings. First, we will consider research question 1, which addresses the practitioners’ inter-team communication tools used. Different practitioners advocate different types of communication tools and we discovered how practitioners tend to adapt to the communication preference of others. Then we consider the second research question. It address the communication at the three studied boundaries, inter team, team and customer, and geographically separated teams.

*RQ1: What inter-team communication tools do practitioners in different roles advocate in our study?*









Concerning the communication means and preference, our findings show that employees were divided into two groups. Developers on one hand preferred slack and recognized the importance of Trello boards and face-to-face. On the other hand, designers, sales, HR, and PR preferred face-to-face communication while recognizing the importance of Trello boards and using Slack when communicating with developers. This reveals how people tend to adapt and improvise ways in order to enhance communication. A summary of communication tools preference is portrayed in table 3; while table 4 presents a summary of communication tools used when practitioners adapt other practitioners’ preferred form of communication. Notice the increase in communication means.




**Table 3. Communication tools preference**

	Developers	Non-Developers
Developers	 	
Non-Developers		 

Legend		
Slack	Trello Board	Face-to-face
		

**Table 4. Communication tools preference after adaption of additional tools**

	Developers	Non-Developers
Developers	 	 
Non-Developers	 	 

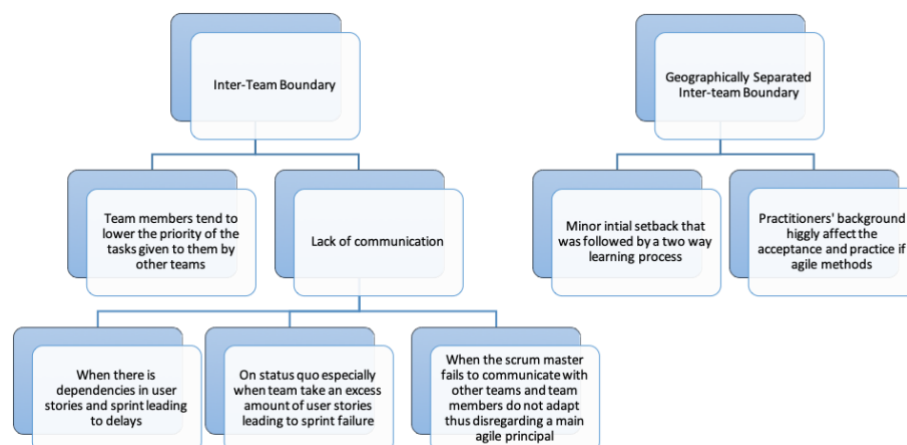
Legend		
Slack	Trello Board	Face-to-face
		

Furthermore, the usage of virtual Kanban boards was agreed upon by all participants. They highlighted the benefits of virtual Kanban boards as means to share knowledge and information while updating user story status without disturbing the flow of work. Knowledge sharing improves the performance of team members and the success of the overall project. This in turns increases the satisfaction of team members and consequently the mutual trust. The relation between knowledge sharing and trust is reciprocal; when one increases the other increases and vice versa.

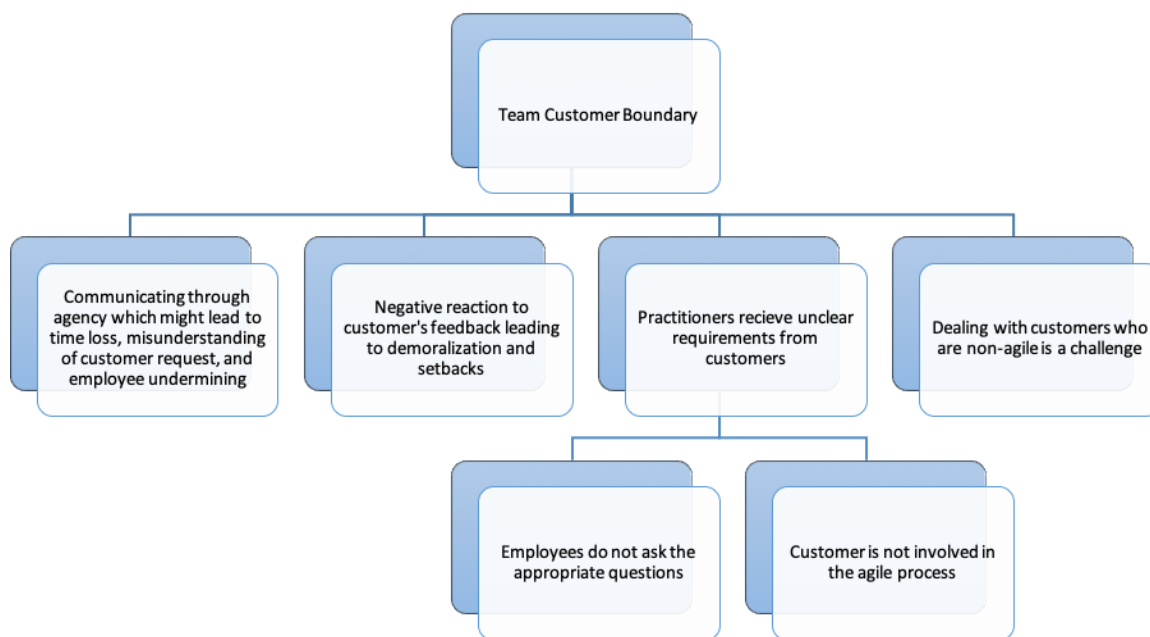
In addition, we observed how developers are already profiting from Slack's features (tagging, pinning, and notifications), while endorsing its usage in specific cases such as short clarifications, information storing platform in case of employee unavailability, and public information broadcasting. On the other hand, HR, PR, sales and designers, surprisingly, do not advocate the usage of Slack as a communication platform, but feel compelled to use it as mean to communicate effectively with developers. Our findings are in agreement with Stray et al.'s findings (Stray, Moe, & Noroozi, 2019). Some users are very active on Slack, while others tend to post fewer messages, adding to an unbalanced activity in which senior developers and technical leaders were most active on Slack. Developers use Slack frequently, while other non-developers adhere to Slack usage in order to communicate with the developers. This leads to an increase in personal mode of coordination and private messages on Slack, instead of public messages (Stray, Moe, & Noroozi, 2019). Scrum masters should stress this importance and encourage the open channel communication between teams. Collaboration and trust in agile projects is built through openness and transparency. Scrum masters should also motivate information sharing and networking behavior. This is done by offering communication skills training (Šmite et al., 2017), giving cross-team effective feedback and encouraging ad-hoc conversations.

*RQ2: What communication bottlenecks do practitioners face at the three studied boundaries, inter-teams, teams and customer, and geographically separated teams?*

Our study imparts the practitioners' communication at the boundaries at three different levels: inter-team, team and customer, and geographically separated inter-team boundaries. We observed problems that rise at the boundaries, while highlighting clarifications to enhance communication at these different boundaries. The figures below highlight the bottlenecks experienced and their respective results at the inter-team and geographically separated inter-team boundaries (Figure 3) and team customer boundary (Figure 4).



**Figure 3. Bottlenecks at inter-team and geographically separated inter-team boundaries**



**Figure 4. Bottlenecks at team customer boundary**

Smooth communication at the boundaries is hard to achieve (Doolin, 2012; Yoo, 2008). At the level of inter- team boundary, dependencies between user stories and lack of communication lead to sprint delays and disruption of the workflow. Delays also occur when a team takes an overload of user stories. In addition to applying the communication tools-mix provided earlier, team members may consider enhancing communication, especially when user stories are interconnected, dependent or on the verge of failure. Our findings align with Lucan et al. (Lucassen, Dalpiaz, van der Werf, Jan Martijn EM, & Brinkkemper, 2015) on the importance of highlighting the dependencies in user stories, particularly when the user story has a non-obvious dependency. These implicit dependencies between user stories should be turned into explicit ones, such as highlighting dependent user stories through storytags that capture the common relevant content, and ensuring the scrum masters involved conduct ad-hoc meeting designed for updates.

At the level of the geographically separated team boundary, both novice and mature teams require time to adjust to the agile methods. Some practitioners weren't even able to adapt to agile methods. The finding highlights an imbalance in both communication and agile implementation between the two sites. The more experienced site (The Netherlands) was able to adapt more than the less experienced site (Kenya). These finding agree with both Šmite et al. (Šmite et al., 2017) and Stray et al. (Stray et al., 2019). Unfamiliarity with tasks and lack of agile practice knowledge are counterpartyed by enhancing communication and motivating particular networking behavior (Šmite et al., 2017). In addition, we observed that team members, having different levels of experience, exchange ideas that lead to mutual benefits and knowledge.

Our findings partially agree with (Moe & Šmite, 2007) which considers face-to-face communication an irreplaceable mean to build trust. Face-to-face communication is essential to build trust especially in the initial stages of the project. Members of the two geographically separated teams met in Kenya. There, the Netherlands team informed the Kenyans on agile software implementations. Later, face-to-face meetings was no longer a

necessity. The follow ups between the two geographically separated teams was done through computer-mediated communication such as instant messaging and video conferencing.

Figure 4 shows customer involvement and feedback is an essential process of agile software development. At the team customer boundary, four main issues were detected. All these challenges lead to decrease in productivity. Our findings agree with (Hoda et al., 2013); not all practitioners have the ability to receive feedback in openness. Negative feedback is common in the agile process since the customer is integrated in the sprint, and feedback is regularly accessible. Such feedback should neither demoralize nor discourage employees, and practitioners should rather understand that customer involvement is in the core of agile. This encouragement is embedded in the form of constructive criticism provided by the product owner in the retrospective. Furthermore, figure 4 shows how challenges rise on the interface between non-agile and agile-implementing companies (such as the company involved in the study) due to fundamental differences. This specially rises if these non-agile implementing companies work in a regulated environment such as the healthcare or governmental sector leading to non-functional requirements, a description of how the software should perform in relation to usability, maintainability, security, and availability. Customers aren't fully aware of all non-functional requirements in the initial stages of the project, instead they focus on fully developing functional requirements. Simultaneously, the agile methodology doesn't provide a clear approach for handling non-functional requirements. An approach detected in this study is turning these non-functional requirements into user stories, making them agile software development tasks. Also, challenges rise when a third party agency is involved. When communication is triangulated, information is lost, altered, or misunderstood. Practitioners should learn to ask specific questions to the customers, especially when the requirements are vague. Further research may involve creating general guidelines for customer requirement clarification.

### *Implications for Practice*

This section provides implications for practitioners to implement in order to enhance communication between the different stakeholders involved in agile software development process. Self-organizing teams and product owners should learn how increase the effective information flow and use the available tools more efficiently. This may be achieved by training product owners and team members on communication skills and raising awareness on the different uses of communication tools available. In addition, teams may regularly use retrospectives to focus on how to transfer rich and concise communication. Make the specification-by-example technique a habit for developers, product owners, and scrum masters. Specification-by-example aids in conversation and highlights the areas for improvement.

Team members should learn how to enhance communication through benefiting from means available and adapting to the usage of certain tools they are not familiar with for the benefit of the whole. For instance, designers aim at using slack for the purpose of facilitating communication with developers. The dependencies between user stories should be highlighted from the beginning through turning the implicit dependencies into explicit ones. Interdependent user stories may be highlighted on virtual Kanban boards to ensure information exchange on the status-quo, which is achieved through encouraging ad-hoc meetings between scrum masters. Lack of agile experience, knowledge and familiarity in



agile methods amongst the less experienced teams can be counterpartyed by encouraging networking behavior and communication. Geographically separated teams may use their differences in their favor in order to benefit and learn from each other's experiences.

In addition, boundary spanners play an important role in reducing the gap between different actors. Finally, when faced with non-functional requirements, practitioners may consider developing them into user stories thus transforming them into agile software development tasks.

## LIMITATIONS

Standards of research quality have been achieved through realizing representativeness and reproducibility of findings, rigor of methods, and generalizability of findings through confirmability, dependability, internal consistency and transferability (Lincoln, 1985).

We achieved confirmability since our conditions and subjects of study follow a grounded theory. The conclusions deduced depend on the practitioner's perception and not the authors' insights. In the study, open coding, category classification, constant comparison and memoing have been described in detail by showing the sequence of conducted steps. All data, including research site description, practitioner interviews' audio and transcribed data were treated anonymously to eliminate any research bias.

Dependability of our conclusions is achieved by collecting data from multiple respondents in two different sites to avoid bias. Our access to data was limited to one company. The company under study is spread across two geographical locations, The Netherlands and Kenya, presenting diversity in interviewed practitioners. Research sites have been anonymized to avoid any exploitation of data for marketing or advertisement purposes. Our study is an exploratory one in an agile software development context on the topics of inter-team communication tools and communication at the three chosen boundaries. We acknowledge that having a single case is a limitation to our study.

However, we have provided a detailed description of the context under study, in the methodology section. This will aid in generalizing the results in future studies. The communication across different boundaries and inter-team communication differs from case to case but analytical induction facilitates the future generalizability of results. Our explanation of the context makes it easier for researchers and practitioners to compare the studied context with their own.

Internal consistency is attained by first ensuring that research sites implement agile. The chosen research site is committed to implement agile as means to provide business solutions for its customers. Interviews with practitioners were also kept private, thus attaining internal credibility by not allowing any other company employee to attend the interview or listen to the recordings.

Transferability is reached by collecting data from two agile-implementing geographical locations. This diverseness allows us to present the preference of communication tools, variance in agile implementation experience, and communication at the boundaries.

---

## CONCLUSIONS

This study follows a qualitative approach to observe the practitioners' preferred form of communication, analyze inter-team communication tools at different boundaries and investigate the difference in challenges faced when implementing agile in different geographical locations. The research site is an agile software development company spread across two geographical location: The Netherlands and Kenya. Participants in the study involved 9 practitioners of different roles and responsibilities.

The study follows the grounded theory approach where open ended interviews were conducted and later performed line-by-line coding, memoing, and constant comparison to the data.

The contribution of the paper is the discovery of the different mechanisms used to overcome the challenges highlighted at the level of the three studied boundaries, inter-teams, teams and customer, and geographically distributed teams. This paper also detects the communication improvement when different preferred communication mechanism between entities are adopted. The findings observed how the preference of the three main inter-team communication means differs among practitioners. Developers tend to prefer the use of while non-developers preferred face-to-face communication. We discovered the different mechanisms to overcome the challenges faced at the boundaries and during day to day communication. Team members should benefit from means available and adapt to the usage of certain tools they are not familiar with. The implicit dependencies between user stories should be tuned into explicit ones and highlighted on virtual Kanban boards. In addition, boundary spanners play an important role in reducing the gap between different actors. Finally, when faced with non-functional requirements should apply agile methods to overcome them.

Recognizing the advancement of software development in the current world, future research may focus on creating innovative methods for knowledge sharing. These methods will aim at decreasing barriers and increase performance outcome. Also further research may involve the development of general guidelines for customer requirements generation when dealing with non-agile implementing customers.

## REFERENCES

Adolph, S., Hall, W., & Kruchten, P. (2011). Using grounded theory to study the experience of software development. *Empirical Software Engineering*, 16(4), 487-513. doi:10.1007/s10664-010-9152-6

Alahyari, H., Berntsson Svensson, R., & Gorschek, T. (2017). *A study of value in agile software development organizations* doi:[https://doi-org.salford.idm.oclc.org/10.1016/j.jss.2016.12.007](https://doi.org.salford.idm.oclc.org/10.1016/j.jss.2016.12.007)

Anika Rani, Shahper Vodanovish, David Sundaram. (2015). Ubiquitous decision making and Support: A framework and evaluation. Paper presented at the *European, Mediterranean & Middle Eastern Conference on Information Systems 2015*, Athens, Greece.

Bass, J. M. (2015). How product owner teams scale agile methods to large distributed enterprises. *Empirical Software Engineering*, 20(6), 1525-1557.

Bass, J. M., Allison, I. K., & Banerjee, U. (2013). Agile method tailoring in a CMMI level 5 organization. *Journal of International Technology and Information Management*, 22(4), 5.

Bass, J. M., & Haxby, A. (2019). Tailoring product ownership in large-scale agile projects: Managing scale, distance, and governance. *IEEE Software*, 36(2), 58-63.

Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., . . . Jeffries, R. (2001). Manifesto for agile software development.

Boden, A., Avram, G., Bannon, L., & Wulf, V. (2009). Knowledge management in distributed software development teams-does culture matter? Paper presented at the *Global Software Engineering, 2009. ICGSE 2009. Fourth IEEE International Conference On*, 18-27.

Boschetti, M. A., Golfarelli, M., Rizzi, S., & Turrinchia, E. (2014). A lagrangian heuristic for sprint planning in agile software development. *Computers and Operations Research*, 43, 116-128. doi:10.1016/j.cor.2013.09.007

Chau, T., & Maurer, F. (2004). Knowledge sharing in agile software teams. *Logic versus approximation* (pp. 173-183) Springer.

CollabNet, a., & Version One, a. (2018). *The 12th annual state of agile report: Long term survey*. Atlanta: VersionOne.

Dingsøy, T., Moe, N. B., Tonelli, R., Counsell, S., Gencel, C., & Petersen, K. (2014). *Agile methods. large-scale development, refactoring, testing, and estimation: XP 2014 international workshops, rome, italy, may 26-30, 2014, revised selected papers* Springer.

Dingsøy, T., Moe, N., Fægri, T., & Seim, E. (2018). Exploring software development at the very large-scale: A revelatory case study and research agenda for agile method adaptation. *Empirical Software Engineering*, 23(1), 490-520. doi:10.1007/s10664-017-9524-2

Dingsøy, T., Nerur, S., Balijepally, V., & Moe, N. B. (2012). *A decade of agile methodologies: Towards explaining agile software development* doi:10.1016/j.jss.2012.02.033

Doolin, B. (2012). Sociomateriality and boundary objects in information systems development. *European Journal of Information Systems*, 21(5), 570-586. doi:10.1057/ejis.2012.20

Drury, M., Conboy, K., & Power, K. (2012). Obstacles to decision making in agile software development teams. *The Journal of Systems & Software*, 85(6), 1239-1254. doi:10.1016/j.jss.2012.01.058

---

Dyba, T., & Dingsoyr, T. (2009). What do we know about agile software development? *Software, IEEE*, 26(5) doi:10.1109/MS.2009.145

Gal, U., Lyytinen, K., & Yoo, Y. (2008). The dynamics of IT boundary objects, information infrastructures, and organisational identities: The introduction of 3D modelling technologies into the architecture, engineering, and construction industry. *European Journal of Information Systems*, 17(3), 290-304.

Ghobadi, S., & Mathiassen, L. (2016). Perceived barriers to effective knowledge sharing in agile software teams. *Information Systems Journal*, 26(2), 95-125. doi:10.1111/isj.12053

Glasser, B. (1978). Theoretical sensitivity: Advances in the methodology of grounded theory. Mill Valley, CA: University of California, *Global software and IT; a guide to distributed development, projects, and outsourcing* (2012). . Portland: Ringgold Inc.

Herbsleb, J. D., & Moitra, D. (2001). Global software development. *Software, IEEE*, 18(2), 16-20. doi:10.1109/52.914732

Hoda, R., & Noble, J. (2017). Becoming agile: A grounded theory of agile transitions in practice. Paper presented at the *Proceedings of the 39th International Conference on Software Engineering*, 141-151.

Hoda, R., Noble, J., & Marshall, S. (2010). Using grounded theory to study the human aspects of software engineering. *Human Aspects of Software Engineering*, 5, 26.

Hoda, R., Noble, J., & Marshall, S. (2012). Developing a grounded theory to explain the practices of self-organizing agile teams. *Empirical Software Engineering*, 17(6), 609-639.

Hoda, R., Noble, J., & Marshall, S. (2013). Self-organizing roles on agile software development teams. *Software Engineering, IEEE Transactions On*, 39(3), 422-444. doi:10.1109/TSE.2012.30

Jovanovic, M., Mesquida, A. -, & Mas, A. (2015). *Process improvement with retrospective gaming in agile software development* doi:10.1007/978-3-319-24647-5\_23

Krehbiel, T. C., & Miller, D. P. (2018). Should agile be part of your quality management system? *Journal of International Technology and Information Management*, 27(1), 2-28.

Licorish, S. A., & MacDonell, S. G. (2014). Understanding the attitudes, knowledge sharing behaviors and task performance of core developers: A longitudinal study. *Information and Software Technology*, 56(12), 1578-1596.

Lincoln, Y. S. (1985). Naturalistic inquiry. *The Blackwell Encyclopedia of Sociology*.

Lucassen, G., Dalpiaz, F., van der Werf, Jan Martijn EM, & Brinkkemper, S. (2015). Forging high-quality user stories: Towards a discipline for agile requirements. Paper presented at the *2015 IEEE 23rd International Requirements Engineering Conference (RE)*, 126-135.

McHugh, O., Conboy, K., & Lang, M. (2011). Agile practices: The impact on trust in software project teams. *IEEE Software*, 29(3), 71-76.

Moe, N. B., & Šmite, D. (2007). *Understanding lacking trust in global software teams: A multi-case study*. Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-540-73460-4\_6

Okhuysen, G. A., & Bechky, B. A. (2009). 10 coordination in organizations: An integrative perspective. *Academy of Management Annals*, 3(1), 463-502.

Patton, M. Q., 1945-. (2002). *Qualitative research and evaluation methods* (3rd ed. ed.). London: London : SAGE.

Petersen, K., Roos, P., Nyström, S., & Runeson, P. (2014). Early identification of bottlenecks in very large scale system of systems software development. *Journal of Software: Evolution and Process*, 26(12), 1150-1171. doi:10.1002/smr.1653

Poppendieck, M. (. B. ). (2003). In Poppendieck T. D., Schwaber K.(Eds.), *Lean software development : An agile toolkit*. Boston Mass.] ; London: Boston Mass. ; London : Addison-Wesley.

Praby, R., & Roland, W. (2016). Can agile processes prevent spectacular information systems development failures. Paper presented at the *European, Mediterranean & Middle Eastern Conference on Information Systems EMCIS 2016*, Krakow Poland.

Rajagopalan, S., & Mathew, S. K. (2016). Choice of agile methodologies in software development: A vendor perspective. *Journal of International Technology and Information Management*, 25(1), 3.

Santos, V., Goldman, A., & de Souza, C. (2015). Fostering effective inter-team knowledge sharing in agile software development. *Empirical Software Engineering*, 20(4), 1006-1051. doi:10.1007/s10664-014-9307-y

Seaman, C. B. (1999). Qualitative methods in empirical studies of software engineering. *IEEE Transactions on Software Engineering*, 25(4), 557-572.

Šmite, D., Moe, N. B., Šablīs, A., & Wohlin, C. (2017). Software teams and their knowledge networks in large-scale software development. *Information and Software Technology*, 86, 71-86.

Šmite, D., Moe, N. B., Šablīs, A., & Wohlin, C. (2017). Software teams and their knowledge networks in large-scale software development. *Information and Software Technology*, 86, 71-86. doi:10.1016/j.infsof.2017.01.003

Stray, V., Moe, N. B., & Noroozi, M. (2019). Slack me if you can!: Using enterprise social networking tools in virtual agile teams. Paper presented at the *Proceedings of the 14th International Conference on Global Software Engineering*, 101-111.

Stray, V., Sjøberg, D. I. K., & Dybå, T. (2016). The daily stand-up meeting: A grounded theory study. *The Journal of Systems & Software, 114*, 101-124. doi:10.1016/j.jss.2016.01.004

Trainer, E. H., & Redmiles, D. F. (2018). Bridging the gap between awareness and trust in globally distributed software teams. *The Journal of Systems & Software, 144*, 328-341. doi:10.1016/j.jss.2018.06.028

Vaivio, J. (2012). *Interviews – learning the craft of qualitative research interviewing* Routledge. doi:10.1080/09638180.2012.675165

Yoo, Y. (2008). The dynamics of IT boundary objects, information infrastructures, and organisational identities: The introduction of 3D modelling technologies into the architecture, engineering, and construction industry. *European Journal of Information Systems, 17*(3), 290-304. doi:10.1057/ejis.2008.13

## APPENDIX A: INTERVIEW GUIDE

I want to ask you about your experience in agile software development. This research involves interviews with different members from the institution you belong to with different roles and responsibilities. I want to learn more about your views on agile software development. Particularly, I am interested in knowing if, and how, tailoring of agile software development methods may enhance the process of software development.

I want to ask you the following questions and tape record your answers. I will keep your responses completely confidential and nothing will be shared with the company. I plan to publish interview extracts but I will make sure your names remain anonymous. Can I switch on the recorder?

### Your Current Project

- How many projects are you working on currently? (If more than one project, ask about each project separately.)
- What is the title of your current project?
- What is your main role in the project?
- What is the overall project structure? And does it differ from one project to another? Questions on the current project:

In your project, may you specify the following?

- What are the main processes?
- Who are the main actors?
- What is the chronological order of processes performed?
- How is the written communication (formal) transferred? And how frequent?
- How and how frequent is informal communication transferred?

Are there any different kinds of ways to communicate other than the ones mentioned?

- How to you send and receive information?
- How do you communicate with your colleges in the team?

- 
- How do you communicate with the people outside the team and at what frequency?
  - How do you measure the size of the software (poking) to be developed?
  - How do you estimate the timeline needed to finish a software development? – What are the factors that you take into consideration?
  - What agile software development disciplines aid in estimating the schedule of the project?
  - What are the factors that negatively affect your delivery with respect to time?
  - How do you list the risks for the project? And at what intervals of time?
  - How often is the customer's feedback portrayed back to you? Does this affect your performance in the next software development project?
  - How do dependencies between teams affect your flow of work?

Do you apply 100 percent agile methods or do you tailor these methods according to your needs? if yes, how? About the interviewee

Again, I will keep your responses completely confidential and nothing will be shared with the company. I plan to publish interview extracts but I will make sure your name remain anonymous. I need the following information just to keep record of the data collected.

- Name:
- Age:
- Organization and Location:
- Position in organization:
- Role in agile applications and techniques:

May you tell me about your qualifications and work experience (in agile).